

PATENT
Case No. N0099US

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
APPLICATION FOR UNITED STATES LETTERS PATENT

INVENTOR: XIANGHENG YANG

TITLE: METHOD FOR DETERMINING THE
INTERSECTION OF POLYGONS USED
TO REPRESENT GEOGRAPHIC
FEATURES

ATTORNEYS: Frank J. Kozak
Lawrence M. Kaplan
NAVIGATION TECHNOLOGIES
CORPORATION
Chicago, Illinois 60654
(312) 894-7000 x7371

09299 0844
T04T30 46E2660

1 METHOD FOR DETERMINING THE INTERSECTION
2 OF POLYGONS USED TO REPRESENT GEOGRAPHIC FEATURES
3

4 BACKGROUND OF THE INVENTION

5 The present invention relates to a method of determining the intersection of two
6 polygons, particularly with respect to determining the intersection of two polygons that
7 represent geographic features.

8 In geographic databases, polygons are used to represent two-dimensional areas,
9 such as lakes, recreational parks, cities, counties, states, golf courses, and so on. A
10 commonly needed function performed using a geographic database is to find that part of
11 one two-dimensional geographic feature (such as a lake, a recreational park, etc.) which
12 is located inside another two-dimensional geographic feature (such as a city, county,
13 state, and so on). The solution can be found by determining the intersection of the
14 polygons used to represent the two-dimensional geographic features.

15 There are prior methods for determining the intersection of polygons. One well
16 known algorithm for determining the intersection of polygons is described in
17 Computational Geometry (O'Rourke, Joseph. Second Edition. Cambridge University
18 Press, 1998). This algorithm takes all shape points from two polygons, sorts them by
19 their y coordinate value and sweep searches the line segments at each shape point from
20 top to bottom. If a line segment is located in both polygons, then it is an intersection
21 boundary. With this prior algorithm, if there are n shape points in two polygons, the
22 overall complexity is $n \cdot \log(n)$. Thus, although this prior algorithm works satisfactorily,
23 it is computationally intensive and accordingly takes a relatively large amount of time to
24 obtain a result.

25 Therefore, there is a need for faster way to determine the intersection of two
26 polygons.
27

SUMMARY OF THE INVENTION

To address these and other objectives, the present invention comprises a method for determining a polygonal intersection of a first polygon and a second polygon. An intersection of the boundary of the first polygon with the boundary of the second polygon is located by finding a node from which at least three portions (i.e., links) of polygon boundaries extend. From this node, a first portion of the boundary of the polygonal intersection is determined by identifying a portion of the boundary of the first polygon that is located inside the second polygon. Each subsequent portion of the boundary of the polygonal intersection is determined by selecting that portion of the boundary of either the first polygon or the second polygon that (1) connects to a leading end of a current portion of the boundary of the polygonal intersection and (2) forms the least angle with the current portion of the boundary of the polygonal intersection.

According to a further aspect of the disclosed invention, intersected holes or islands are determined if they are in both polygons.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 is a diagram that illustrates a geographic region and data representations of some of the features located in the geographic region.

Figure 2 is a diagram that illustrates a polygon used to represent a two-dimensional geographic area and a data representation of the polygon.

Figure 3 is a block diagram illustrating a software program that determines the intersection of two polygons.

Figure 4 depicts two intersecting polygons and illustrates operation of a process performed by the software program of Figure 3.

Figures 5A-5C are a flow chart of the process performed by the software program of Figure 3.

Figure 6 depicts a step in the process of Figures 5A-5C.

Figures 7A and 7B depict another step in the process of Figures 5A-5C.

Figure 8 depicts another step in the process of Figures 5A-5C.

Figure 9 shows a polygon with a hole and data representation thereof.

Figure 10 shows an intersection of polygons with holes in the intersection.

DETAILED DESCRIPTION OF THE PRESENTLY PREFERRED EMBODIMENTS

I. THE GEOGRAPHIC DATABASE

Referring to Figure 1, a geographic database 100 includes data that represent features that are located in a covered region 108. For example, the geographic database 100 may contain data that represent roads 112 and points of interest 116. The geographic database 100 also includes data 120 that represent two dimensional features or entities 124. Some of the different kinds of two dimensional features that can be represented in the geographic database 100 include land features (e.g., mountain ranges), bodies of water (e.g., lakes, ponds, swamps, wetlands), recreation areas (e.g., golf courses, parks, stadiums, forest preserves), business parks, industrial parks, and governmental administrative areas (e.g., cities, towns, states, townships, municipalities, counties, school districts), as well as other types of areas.

Referring to Figure 2, in the geographic database 100, two-dimensional features are represented using polygons. In other words, in the geographic database 100, the data that are used to represent two-dimensional features describe these features as polygons. In the geographic database 100, the boundary of the polygon corresponds to the boundary of the represented two-dimensional feature.

The following terminology applies to the representation of two-dimensional features in the geographic database 100.

“Node” - A point that terminates a link.

“Line segment” - A straight line connecting two points.

“Link” (or “edge”) - A contiguous, non-branching string of one or more line segments terminating in a node at each end.

“Shape point” - A point along a link between two nodes.

“Oriented Link” - A link that has a starting node (referred to as the “reference node”) and an ending node (referred to as the “non reference node”).

“Simple Polygon” - An interior area of an outer boundary formed by a string of oriented links that begins and ends in one node. (A simple polygon does not cross itself.)

1 “Polygon” - The area bounded by an outer boundary and none or at least one
 2 interior boundary (e.g., a hole or island). A polygon is constructed from one outer simple
 3 polygon and none or at least one inner simple polygon. A polygon is simple if it just
 4 consists of one simple polygon, or complex if it has at least one inner simple polygon.

5 The geographic database 100 follows certain conventions. For example, links do
 6 not cross themselves and not cross each other except at a node. Also, there are no
 7 duplicated shape points, nodes or links. Two links that connect to each other have a
 8 common node. In the geographic database 100, overlapping two-dimensional geographic
 9 features are represented by overlapping polygons. When polygons overlap, the boundary
 10 of one polygon crosses the boundary of the other polygon. In the geographic database
 11 100, the location at which the boundary of one polygon intersects the boundary of
 12 another polygon is represented by a node. (A node may be used to represent other
 13 locations along the boundary of a polygon other than a location at which the boundary of
 14 the polygon intersects the boundary of another polygon.) A shape point is not used to
 15 represent a point at which the boundary of a polygon intersects the boundary of another
 16 polygon.

17 In the geographic database 100, a data representation 132 of a polygon that
 18 represents a two-dimensional feature includes a list 136 that identifies each oriented link
 19 that forms the boundary of the polygon. According to a convention used in the
 20 geographic database, the list 136 identifies the oriented links in clockwise order. If the
 21 polygon has a hole (i.e., a second polygon located entirely inside the first polygon), the
 22 hole is identified by a list of oriented links listed in counter clockwise order.

23 24 II. METHOD FOR DETERMINING THE INTERSECTION OF POLYGONS

25 As mentioned above, with applications that use geographic databases, there is
 26 sometimes a need to determine whether two two-dimensional features overlap and, if
 27 they do, to determine the two-dimensional area that forms the intersection (i.e., the
 28 overlapping portion) of the two two-dimensional areas. In terms of the polygons used to
 29 represent these two-dimensional features, there is a need to determine the intersection of
 30 polygons.

1 A method for determining the intersection of polygons is described in connection
2 with Figures 3-10. Referring to Figure 3, the method is implemented by a software
3 program 200 that uses the geographic database 100 that contains data representations of
4 polygons, as described in connection with Figures 1 and 2. The software program 200
5 receives two data representations of polygons as an input. (The two data representations
6 of polygons conform to those described in Figure 2.) The software program 200
7 provides, as an output, a data representation of each polygon that represents a polygonal
8 intersection of the two data representations of polygons that were provided as input.
9 Alternatively, the software program 200 provides an output indicating that there is no
10 intersection of the two data representations of polygons that were provided as input.

11 The software program 200 may be installed on a navigation system that uses a
12 geographic database to provide navigation-related features to a driver of a vehicle. The
13 software program 200 may also be installed on a server that uses a geographic database to
14 provide navigation-related services to various types of users, including vehicle drivers,
15 persons who are not drivers, users of personal digital assistants (PDAs), cell phone users,
16 and so on. The server that uses the geographic database may be connected to the Internet
17 or other communications network. The software program 200 may also be used when
18 compiling a geographic database, i.e., to produce a derived database product from a
19 source database. The software program 200 may also be used on a standalone computer
20 that uses a geographic database to provide various navigation-related, map-related, or
21 other types of features.

22 In any of these different systems, the software program 200 may be stored on a
23 computer-readable medium, loaded from the medium into a memory of the system and
24 run, as needed.

25 The determination of the polygonal intersection of two polygons is described with
26 reference to Figures 4-10. Figure 4 shows two polygons, labeled P(A) and P(B). Using
27 the disclosed embodiment, the intersection, P(I), of polygons P(A) and P(B) is
28 determined.

29 Figures 5A-5C shows steps in a process 220 performed by the software program
30 200 for determining the intersections of the polygons P(A) and P(B). The process 220

1 begins with a step in which the data representations of the two polygons are received
 2 (Step 224). The process 220 includes a step that forms minimum bounding rectangles for
 3 the two polygons P(A) and P(B) (Step 228). Figure 6 depicts formation of these
 4 minimum bounding rectangles for P(A) and P(B), labeled MBR(A) and MBR(B),
 5 respectively. The minimum bounding rectangles, MBR(A) and MBR(B), are compared
 6 to determine whether they intersect (Step 232). If the minimum bounding rectangles of
 7 P(A) and P(B) do not intersect, then the polygons P(A) and P(B) do not intersect, and a
 8 negative result is returned (Step 236). If the minimum bounding rectangles MBR(A) and
 9 MBR(B) intersect, then the polygons P(A) and P(B) may intersect and the process 220
 10 continues.

11 The process continues by forming a node-link map. The node-link map is formed
 12 using the polygonal intersection, MBR(A_B) of the minimum bounding rectangles
 13 MBR(A) and MBR(B) of the polygons P(A) and P(B). This polygonal intersection
 14 P(A_B) is determined as shown in Figure 6. To form the node-link map, each link of
 15 either polygon, P(A) or P(B), which is located entirely inside the polygonal intersection
 16 P(A_B) of the minimum bounding rectangles MBR(A) and MBR(B) of the polygons
 17 P(A) and P(B) is identified (Step 244). (A link is located entirely inside the intersection
 18 MBR(A_B) of the minimum bounding rectangles MBR(A) and MBR(B) if both ends of
 19 the link are inside the intersection MBR(A_B) of the minimum bounding rectangles
 20 MBR(A) and MBR(B).) Then, each node of each of these identified links is identified.
 21 (Shape points can be ignored at this step.) Each of these identified nodes is then included
 22 as an entry in the node-link map (Step 248). For each node included as an entry in the
 23 node-link map, all the links that connect to the node are identified and associated with the
 24 node in the map.

25 The node-link map is used to identify a node (referred to herein as "N(START)")
 26 that is located on the polygon P(I) that forms the intersection of the input polygons P(A)
 27 and P(B). Using the node-link map, a node is identified that has three (or more) links
 28 associated with it (Step 252). A node from the node-link map that has three (or more)
 29 links associated with it is located on the polygon P(I) that forms the intersection of the
 30 two polygons P(A) and P(B). Because this node is located on the polygon P(I) that forms

1 the intersection of the two polygons P(A) and P(B), two (and only two) of the three or
 2 more links that connect to the node N(START) are links of the intersection polygon P(I).
 3 The next step is to identify which two of the three or more links form part of the
 4 intersection polygon P(I).

5 In order to identify which of the links that connect to N(START) form part of the
 6 intersection polygon P(I), a process is performed that identifies one link of one of the
 7 original polygons, P(A) or P(B), that is located inside the other of the original polygons,
 8 P(B) or P(A) (Step 254). A link of one of the original polygons that is located inside the
 9 other of the original polygons is part of the intersection polygon P(I). This link is
 10 referred to as "LINK(START)." Details of the process for finding this link are illustrated
 11 in Figures 7A, 7B and 8.

12 Referring to Figures 7A and 7B, in P(A) and P(B), all edges are oriented
 13 clockwise. Node N is the starting node since four edges are connected to it, i.e., ae1 and
 14 ae2 from P(A), be1 and be2 from P(B). If ae2 is on the right side of be1 and be2, then
 15 ae2 is inside P(B) and can be selected as the starting edge of the polygon intersection (as
 16 shown in Figure 7A). If ae1 is on the right side of be1 and be2, then ae1 is inside P(B)
 17 and can be selected as the starting edge of the polygon intersection. If ae2 and ae1 both
 18 are not on the right side of be1 and be2, there is no intersection at N (as shown in
 19 Figure 7B). To determine whether ae2 is on the right side of be1 and be2, the process
 20 500 shown in Figure 8 can be used. Using the process 500 in Figure 8, the shape point
 21 bp1 is selected from be1, the shape point bp2 is selected from be2 and the shape point
 22 ap2 is selected from ae2. These shape points are selected because they are the closest
 23 shape points on their respective edges to node N(point p) although other shape points on
 24 these edges may be selected as well. The process 500 in Figure 8 is used to test point if
 25 ap2 is on the right side of the line segment bp1_p and p_bp2. If so, ae2 is to the right of
 26 be1 and be2 (as shown in Figure 7A). If not, ae2 is not on the right of be1 and be2 (as
 27 shown in Figure 7B).

28 After LINK(START) is identified, the next link that forms the intersection
 29 polygon P(I) is identified. This next link is identified by examining all the links that
 30 connect to the clockwise end of the LINK(START). Referring to Figure 5B, the

1 clockwise end of the LINK(START) is determined from the clockwise direction of the
 2 links in the original polygon (P(A) or P(B)) from which LINK(START) was taken
 3 (Step 264).

4 If there is only one other link that connects to the clockwise end of the link,
 5 LINK(START), then this link is the next link of the intersection polygon P(I). Data
 6 indicating this next link are stored in the list formed to represent the intersection polygon
 7 P(I) (Steps 272 and 276).

8 If there are two or more other links that connect to the clockwise end of the
 9 LINK(START), then the link that forms the smallest angle in the counterclockwise
 10 direction is the next link of the intersection polygon P(I). Data indicating this next link
 11 are stored in the list formed to represent the intersection polygon P(I) (Steps 272, 278 and
 12 276).

13 After this link (i.e., the link after LINK(START)) is added to the list formed to
 14 represent the intersection polygon P(I), the process continues by determining the next
 15 link that forms the intersection polygon P(I). This next link is determined in a similar
 16 manner as just described. That is, all the links that connect to the clockwise end of the
 17 link are identified (Steps 264 and 272, again). If there is only one other link, then this
 18 link is added to the list of links that form the intersection polygon P(I). If there are two or
 19 more other links, then the next link is the one that forms the smallest angle in a
 20 counterclockwise direction from the link most recently identified as being part of the
 21 intersection polygon P(I) (Step 278, again).

22 The process continues until the N(START) is encountered (Step 280). When
 23 N(START) is encountered, then one simple polygon (that forms part or all of the entire
 24 polygon intersection P(I) of P(A) and P(B)) has been identified. Data describing this
 25 simple polygon are contained in the list 258. These data are added to a simple polygon
 26 list in a data representation 318 (in Figure 5C) of the entire intersection polygon P(I) of
 27 P(A) and P(B) (Step 284).

28 All used links and nodes in the link-node map are removed (Step 288). If there
 29 are still nodes with three or more connected links in the node-link map after a simple
 30 polygonal intersection has been identified (Step 292 in Figure 5B), then there is at least

one more distinct simple polygonal intersection between the first polygon and the second polygon. The process is repeated (starting at Step 252 in Figure 5A) to determine each additional simple polygonal intersection. The process is repeated until there is no node in the link-node map that has more than three links connected to it.

The process 220 finds intersected holes and islands in P(A) and P(B). Some two-dimensional features represented in the geographic database 100 are encompassed entirely within other represented two-dimensional features. As an example, an island is located entirely within a body of water. In the geographic database 100, a two-dimensional feature that is encompassed entirely within another two-dimensional is represented as a hole inside a polygon.

The data representation of the polygon containing a hole includes (1) a list of edges that define the boundary of the entire polygon (i.e., similar to Figure 2, described above) and (2) a list of edges that define the boundary of the hole located inside the polygon. According to a convention used in the geographic database 100, the list of edges in the data representation of a hole inside a polygon are ordered in counter-clockwise order. Figure 9 shows a polygon with a hole and data representation thereof.

When determining the intersection of two polygons that have holes, it is possible that some of the holes are included in the intersection. Figure 10 illustrates the intersection of two polygons having holes/islands in their intersection.

When the disclosed process 220 is used to determine the polygonal intersection of two polygons, the data representation of the polygonal intersection includes a representation of any holes of either of the original polygons that are located inside the polygonal intersection. First, a determination is made whether the minimum bounding rectangle of the hole is located inside the intersection of the minimum bounding rectangles of the two original polygons. If the minimum bounding rectangle of the hole is not located completely inside the intersection of the minimum bounding rectangles of the two original polygons, then the hole is not located in the polygonal intersection. If the minimum bounding rectangle of the hole is located inside the intersection of the minimum bounding rectangles of the two original polygons, then a determination is made whether any point (e.g., any node) of the hole is located inside the other of the polygons.

1 Figure 5C shows steps in the process 220 for handling holes/islands in the
2 intersection polygon, P(I). Starting with an island/hole from polygon A, a search is made
3 to find an immediate parent simple polygon in polygon B. If there is a simple polygon in
4 B and its orientation is clockwise, then the island/hole from polygon A is a simple
5 intersection polygon. If there is no parent polygon in B or the orientation of the parent
6 simple polygon in B is counter clockwise, then the island/hole from polygon A is not a
7 simple intersection polygon (Step 302). These steps are repeated until all islands/holes
8 from polygon A have been considered.

9 The process continues by considering an island/hole from polygon B and
10 searching for an immediate parent simple polygon in polygon A. If there is a simple
11 polygon in A and its orientation is clockwise, then the island/hole from polygon B is a
12 simple intersection polygon. If there is no parent polygon in A or the orientation of the
13 parent simple polygon in A is counter clockwise, then the island/hole from polygon B is
14 not a simple intersection polygon (Step 306). This process is repeated until all
15 islands/holes from polygon B have been visited.

16 After accounting for any holes or islands in the intersection, a data representation
17 of the entire complex polygon intersection, P(I), is built from the list(s) of simple
18 polygons list 318 (Step 307). The intersection of P(A) and P(B) may include one or more
19 polygons, each of which may be either simple or complex. The data representation of the
20 polygonal intersection can then used by other applications in the system in which the
21 software program 200 is installed that require data indicating the polygonal intersection
22 of the P(A) and P(B) (Step 288)

23

24 III. ADVANTAGES

25 The present method for determining the intersection of polygons is
26 computationally less intensive than prior methods. The present method achieves this
27 advantage, in part, because only the polygon edge is considered instead of entire line
28 segments, therefore greatly reducing the amount of computation.

29

1 IV. FURTHER CONSIDERATIONS2 Consistent ordering – clockwise or counterclockwise

3 In the embodiments described above, it was explained that the data
4 representations of polygons included lists of links used to represent the boundaries of the
5 polygons and that the links contained on a list were ordered corresponding to a clockwise
6 traversal of the represented polygon boundary via the links. It is not required that the
7 lists of links used to represent polygons identify the links in clockwise order so long as
8 the order in which the links are listed is consistent throughout the geographic database.
9 For example, according to an alternative embodiment, instead of listing the links that
10 form polygon boundaries in a clockwise order, data representations of polygons could list
11 the links in a counterclockwise order. If a counterclockwise order is used for polygon
12 boundaries, it might be preferable to use a clockwise order to represent holes in polygons.

13
14 It is intended that the foregoing detailed description be regarded as illustrative
15 rather than limiting and that it is understood that the following claims including all
16 equivalents are intended to define the scope of the invention.